We Claim:

1. In a program development environment, a method comprising the steps of:

providing a parse tree data structure written in a base language;

defining an assignment function, said assignment function taking a plurality of parse tree structures as arguments; and

calling said assignment function to determine the value of at least one assignment within at least one of said base language and a base language extension to said base language.

2. The method of claim 1, comprising the further steps of:

representing said parse tree data structure as a class, said class being the basis for a plurality of parse tree objects, said parse tree objects including methods able to retrieve values for base language objects;

defining said assignment function in more than one class, said assignment function taking a plurality of parse tree objects as arguments; and

overloading said assignment function.

3. The method of claim 2 wherein said assignment function overloads a mathematical operator.

4. The method of claim 2 wherein the overloading of said assignment function is based on the context of the base language objects.

5. The method of claim 2, comprising the further step of:

evaluating said class at compile-time, and

adjusting the resulting class definitions from said evaluation to increase the efficiency of run-time performance.

6. The method of claim 2, comprising the further step of:

overloading a mathematical operator with said assignment function to alter the sequence of evaluation of operands usually followed in said programming language, said overloading designating the order of operand evaluation.

7. The method of claim 2, comprising the further step of:

calling a method in said parse tree object to determine the type of operator at the root of a tree.

8. The method of claim 2, comprising the further step of:

calling a method in said parse tree object to retrieve one of an associated left and right tree.

9. The method of claim 1 wherein the root of said parse tree data structure is one of a constant, variable, a mathematical symbol and a mathematical expression.

10. The method of claim 1 wherein said assignment function is not explicitly defined.

11. The method of claim 1 wherein said assignment function is used to identify in-place operations.

12. The method of claim 1 wherein said assignment function is used to identify and perform multiply and accumulate ("MAC") operations.

13. The method of claim 1 wherein said base language is one of C++, Java, System-C, VHDL, Verilog, C#, IDL, MATLAB and a language based on the .Net framework.

14. In a program development environment, a method comprising the steps of:

providing a parse tree data structure written in a base language;

defining an assignment function, said assignment function taking a plurality of parse tree structures as arguments;

calling said assignment function to determine the value of at least one assignment within at least one of a base language and a base language extension; and

generating code for an embedded processor using said parse tree data structure.

15. In a program development environment, a method comprising the steps of:

providing a parse tree data structure written in a base language;

defining an assignment function, said assignment function taking a plurality of parse tree structures as arguments;

calling said assignment function to determine the value of at least one assignment within at least one of a base language and a base language extension; and

using said parse tree data structure in software emulation.

16. A medium for use in a program development environment, said medium holding instructions for execution of a method on an electronic device, said method comprising the steps of:

providing a parse tree data structure written in a base language;

defining an assignment function, said assignment function taking a plurality of parse tree structures as arguments; and

calling said assignment function to determine the value of at least one assignment within at least one of said base language and a base language extension to said base language.

17. The medium of claim 16, wherein said method comprises the further steps of:

representing said parse tree data structure as a class, said class being the basis for a plurality of parse tree objects, said parse tree objects including methods able to retrieve values for base language objects;

defining said assignment function in more than one class, said assignment function taking a plurality of parse tree objects as arguments; and

overloading said assignment function.

18. The method of claim 17 wherein said assignment function overloads a mathematical operator.

19. The method of claim 17 wherein the overloading of said assignment function is based on the context of the base language objects.

20. The medium of claim 17, wherein said method comprises the further steps of:

evaluating said class at compile-time, and

adjusting the resulting class definitions from said evaluation to increase the efficiency of run-time performance.

21. The medium of claim 17, wherein said method comprises the further step of:

overloading a mathematical operator with said assignment function to alter the sequence of evaluation of operands usually followed in said programming language, said overloading designating the order of operand evaluation.

22. The medium of claim 17, wherein said method comprises the further step of:

calling a method in said parse tree object to determine the type of operator at the root of a tree.

23. The medium of claim 17, wherein said method comprises the further step of:

calling a method in said parse tree object to retrieve one of an associated left and right tree.

24. The medium of claim 16 wherein the root of said parse tree data structure is one of a constant, variable, a mathematical symbol and a mathematical expression.

25. The medium of claim 16 wherein said assignment function is not explicitly defined.

26. The medium of claim 16 wherein said assignment function is used to identify in-place operations.

27. The medium of claim16 wherein said assignment function is used to identify and perform multiply and accumulate ("MAC") operations.

28. The medium of claim 16 wherein said base language is one of C++, Java, System-C, VHDL, Verilog, C#, IDL, MATLAB and a language based on the .Net framework.

29. The medium of claim 16 wherein said parse tree data structure is used to generate code for an embedded processor.

30. The medium of claim 16 wherein said parse tree data structure is used in processor emulation.

31. In an electronic device, a system, comprising:

a program development environment, said program development environment having a base language;

a parse tree data structure including methods able to retrieve values for base language objects; and

an assignment function taking a plurality of parse tree structures as arguments, said assignment function called to determine the value of at least one assignment within at least one of said base language and a base language extension to said base language.

32. The system of claim 31 wherein said parse tree data structure is represented as a class, said class being the basis for a plurality of parse tree objects, said parse tree objects including methods able to retrieve values for base language objects.

33. The system of claim 32 wherein said assignment function is defined in more than one class.

34. The system of claim 33 wherein said assignment function is overloaded.

35. The system of claim 31 wherein said base language is one of C++, Java, System-C, VHDL, Verilog, C#, IDL, MATLAB and a language based on the .Net framework.

36. The system of claim 31 wherein said parse tree data structure is used to generate code for an embedded processor.

37. A medium for use in a program development environment, said medium holding instructions for execution of a method on an electronic device, said method comprising the steps of:

providing a parse tree data structure written in a base language;

defining an assignment function, said assignment function taking a plurality of parse tree structures as arguments;

calling said assignment function to determine the value of at least one assignment within at least one of said base language and a base language extension to said base language; and

generating code for an embedded processor using said parse tree data structure.

38. A medium for use in a program development environment, said medium holding instructions for execution of a method on an electronic device, said method comprising the steps of:

providing a parse tree data structure written in a base language;

defining an assignment function, said assignment function taking a plurality of parse tree structures as arguments;

calling said assignment function to determine the value of at least one assignment within at least one of said base language and a base language extension to said base language; and

using said parse tree data structure in software emulation.

39. In an object-oriented program development environment having a base language, a method comprising the steps of:

providing a parse tree data structure in said object-oriented program development environment, said parse tree data structure used as the basis for at least one parse tree object, said parse tree objects including methods able to retrieve values for base language objects;

defining an assignment function taking a plurality of parse tree structures as arguments in more than one class;

calling said assignment function to determine the value of at least one assignment within at least one of a base language and a base language extension; and

overloading a mathematical operator with said assignment function based on the context of a plurality of base language objects.